AMENDMENTS TO THE SPECIFICATION

Please amend paragraph 0017 as follows:

[0017] According to one embodiment of the invention, a system utilizes round robin scheduling or a similar scheduling algorithm. The algorithm schedules tasks from multiple queues (or sub-queues within a queue), with each queue representing a particular priority level or levels. If it is determined that there are no more tasks in the highest priority queue, the algorithm pauses the scheduling process to obtain more tasks for scheduling and then resumes scheduling. A particular embodiment includes a delay process. In this embodiment, if a process obtains more tasks for scheduling and determines that there are no tasks in the highest priority queue, then a delay period is commenced. During the delay period, the process does not obtain more tasks when it is determined that there are no tasks in the highest priority queue. The delay period may provide a time period in which scheduling [[is]] continues while time is allowed for more high priority tasks to arrive.

Please amend paragraph 0022 as follows:

[0022] **Figure 2** illustrates an embodiment of queue formation for scheduling. In this illustration, the processes are described in terms of a first thread (thread 1 **205**) that relates to the formation of queues for received packets into a priority packet queue <u>(first queue)</u> **230** and a second thread (thread 2 **210**) that relates to copying packets from the priority packet queue into a copied packet queue <u>(second queue)</u> **250**. Other embodiments may institute the processes in a different manner. While this description refers to ordering packets and copying packets, this may be understood to mean that

pointers or other similar devices are used to reference the packets. The priority packet queue **230** and the copied packet queue **250** each comprise multiple individual queues of packets, which may be referred to as sub-queues. In this example, each individual sub-queue represents one of a number of different priority levels. This structure may vary in different embodiments of the invention. For example, one sub-queue may represent more than one different priority levels.

Please amend paragraph 0025 as follows:

[0025] **Figure 3** illustrates an embodiment of round robin scheduling by a scheduler. In this illustration, an algorithm **305** controls a scheduling operation for packets that are to be provided to, for example, a device driver **330**. The scheduling operation provides for scheduling of packets of multiple different priorities. The packets illustrated are in a priority 1 ~~queue~~ <u>sub-queue</u> **310**, a priority 2 ~~queue~~ <u>sub-queue</u> **315**, and a priority 3 ~~queue~~ <u>sub-queue</u> **320**, although any number of priority levels and types may be present. To illustrate the round robin scheduling operation, a scheduling unit **325** takes one or more packets from priority 1 **310**, one or more packets from priority 2 **315**, and one or more packets from priority 3 **320**. The scheduling unit **325** then begins again with priority 1 **310**, as shown figuratively by the scheduling unit **325** rotating from each priority level to the next level.

Please amend paragraph 0026 as follows:

[0026] In a weighted round robin scheduling operation, the number of packets scheduled for each priority is weighted based on the priorities of the packets. For example, three packets may be obtained from the priority 1 ~~queue~~ <u>sub-queue</u> **310**, two

packets may be obtained from the priority 2 ~~queue~~ sub-queue **315**, and one packet may be obtained from the priority 1 ~~queue~~ sub-queue **320**. The scheduled packets are provided to device driver **330**. In the process of scheduling packets, the packets in the highest priority ~~queue~~ sub-queue, priority 1 ~~queue~~ sub-queue **310**, may be exhausted. According to an embodiment of the invention, if the packets in the highest priority ~~queue~~ sub-queue are exhausted, the algorithm **305** provides for copying additional packets before scheduling is resumed. In another embodiment, if additional packets are copied and there are still no packets in the highest priority ~~queue~~ sub-queue **310**, then a delay period may be commenced. The scheduling then will continue with the available packets without copying any more packets until the delay period has expired. While for simplicity this example only refers to exhausting the highest priority ~~queue~~ sub-queue, other embodiments might provide for copying packets if one or more different priority ~~queues~~ sub-queues are exhausted.

Please amend paragraph 0027 as follows:

[0027] **Figure 4** is a flow chart of an embodiment of adaptive queue scheduling. In this illustration, received packets are copied into a copied packets queue (also referred to herein as the second queue) for scheduling **405**. The copied packets queue includes packets arranged in a number of sub-queues, each sub-queue representing one of a number of different priorities. In this example, three priority levels (P1 - high, P2 - medium, P3 low) are provided, but any number and type of priority levels may be provided. To the degree that packets are available, a certain number of packets are scheduled from each priority level. In a weighted scheduling algorithm, the scheduling of the packets is weighted according to the priority levels of the packets. In this example,

a driver **425** is sent a set of three P1 packets **410**, two P2 packets **415**, and one P3 packet **420** (if this number of packets is available for each priority level) in each scheduling action.

Please amend paragraph 0028 as follows:

[0028] After a set of packets has been scheduled and sent, there is a determination whether all of the sub-queues are empty **435**. If so, then there are no packets available for scheduling, and the process continues with copying received packets into the copied packets queue <u>(second queue)</u> **440**. The process then returns to sending packets from each of the priority sub-queues **410-420**. If all of the priority sub-queues are not empty **435**, there is then a determination whether the high priority sub-queue P1 is empty **445**. If the P1 sub-queue is not empty, then the process of scheduling packets from the priority sub-queues continues **410-420**. However, if the P1 sub-queue is empty, then the process provides for copying received packets into the copied packets queue **440** and proceeding to the process of scheduling packets from the priority sub-queues continues **410-420**. By copying such received packets, the process may avoid scheduling lower priority packets while higher priority packets are not addressed. While the illustrated operations in Figures 4 and 5 make a determination whether any of the highest priority (P1) packets are available in the queue, embodiments of the invention are not limited to this type of example. An embodiment may, for example, make a determination whether any packets of one or more different priority levels are available.

Please amend paragraph 0038 as follows:

[0038] Portions of the present invention may be provided as a computer program product, which may include a ~~machine-readable~~ computer-readable medium having stored thereon instructions, which may be used to program a computer (or other electronic devices) to perform a process according to the present invention. The ~~machine-readable~~ computer-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnet or optical cards, flash memory, or other type of media / ~~machine-readable~~ computer-readable medium suitable for storing electronic instructions. ~~Moreover, the present invention may also be downloaded as a computer program product, wherein the program may be transferred from a remote computer to a requesting computer by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., a modem or network connection).~~